Neurosys Recording Hardware and Software

User's Manual

Contents

Chapter 1: System requirements. 3

Chapter 2: Shipped contents: 4

Chapter 3: Install DAQ board and drivers....7

Chapter 4: Install software 8

Chapter 5: System self-test....11

Chapter 6: Navigating the interface. 16

Chapter 7: Real-time spike sorting....25

Chapter 8: ELECTRODE FABRICATION....27

Chapter 9: Matlab integration 28

Chapter 10: Med-PC® integration....30

Chapter 11: Reading PLX files in MATLAB or C/C++....32

Chapter 1: System requirements.

To use your system, you will need the following items:

1. PC running Windows XP, 7, 8, 10, or 11.
2. One free PCI or PCI express slot.
3. Speakers (plugged into computer's audio port) to listen to neural signals.

Chapter 2: Shipped contents:

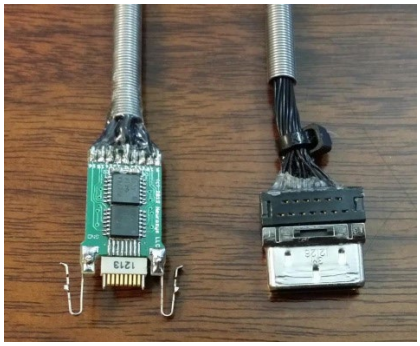
Your system includes the following items:

1. 16-channel amplifier and cables:

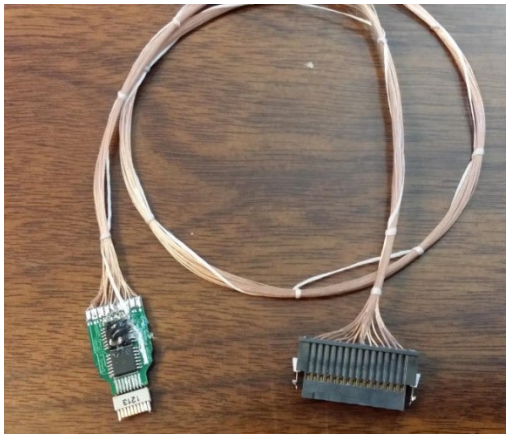
VHDCI to SCSI cable

24V AC adapter

2. Rat systems have headstage with spring-protected cable:



3. Mouse systems have a smaller headstage with an ultra-light cable:

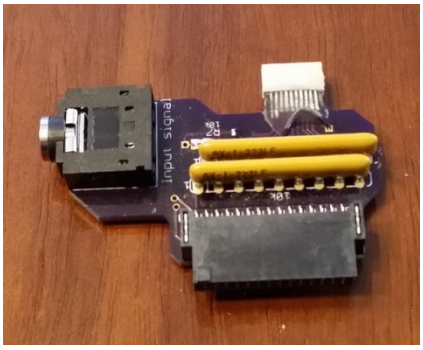


4. DAQ board from National Instruments:

5. Extension cable for preamplifier input, allowing recording chamber to be some distance away from amplifier.

6. Rat systems include a commutator that can be attached to a Med Associates gimbal arm (Med Associates part numbers PHM-115-SAI and FAB-PHM-110C).

7. Self-test board and cable.



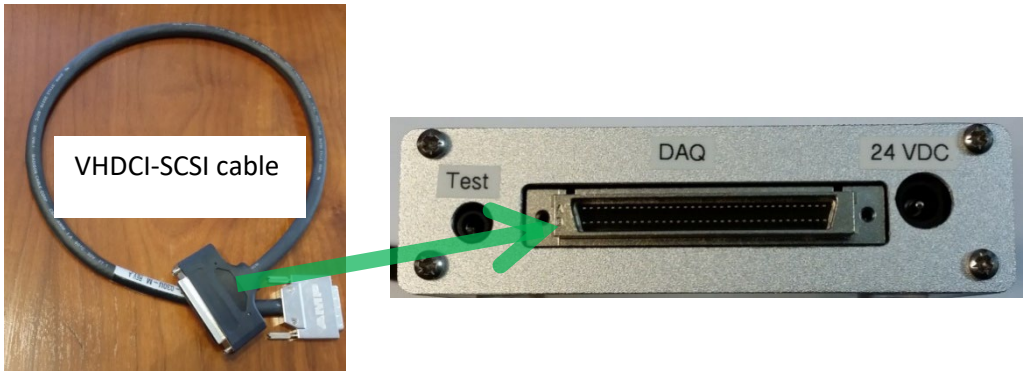
Chapter 3: Install DAQ board and drivers

Please install the National Instruments digital acquisition board ("DAQ board") into an available PCIe slot, then download the latest NI-DAQmx drivers by pasting this link into your browser:

http://search.ni.com/nisearch/app/main/p/bot/no/ap/tech/lang/en/pg/1/sn/catnav:du/q/ni-daqmx%20runtime/

Alternately, you can go to www.ni.com/downloads, search for "`ni-daqmx runtime`", and choose the most recent version.

The download and install will take a while. In the meantime, you can connect the preamplifier to the DAQ card with the VHDCI-SCSI cable, and connect the 24V power adapter:



Once the NI-DAQmx drivers have installed, proceed to software installation (next chapter).

Chapter 4: Install software

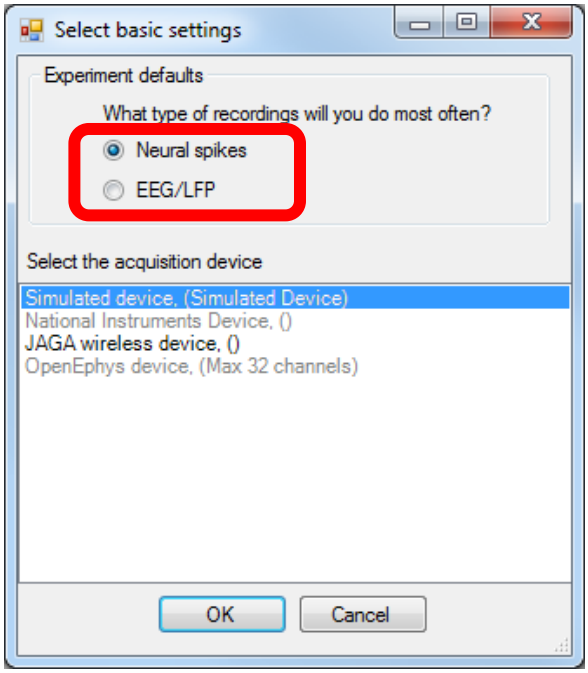
Download and run the installer from www.neurosysllc.com/Software.html. There are two programs: "**NeuroPhys**", which is the main acquisition program, and "**NeuroSorter**", which is an offline analysis program.

Launch NeuroPhys by clicking the "brain" icon on the desktop:

At first launch, NeuroPhys will ask you to register the product:

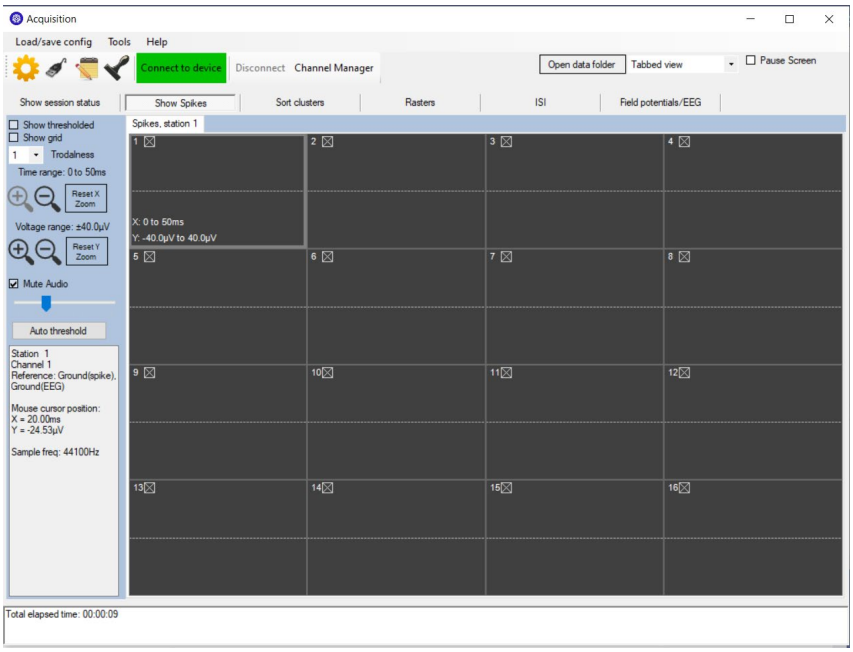
Click "Register using XML file", and select the .XML license file you were sent. If you do not want to register, you can run a FREE version that is **fully functional forever**, but with only 4 active channels.

When first run, the following configuration dialog will appear:

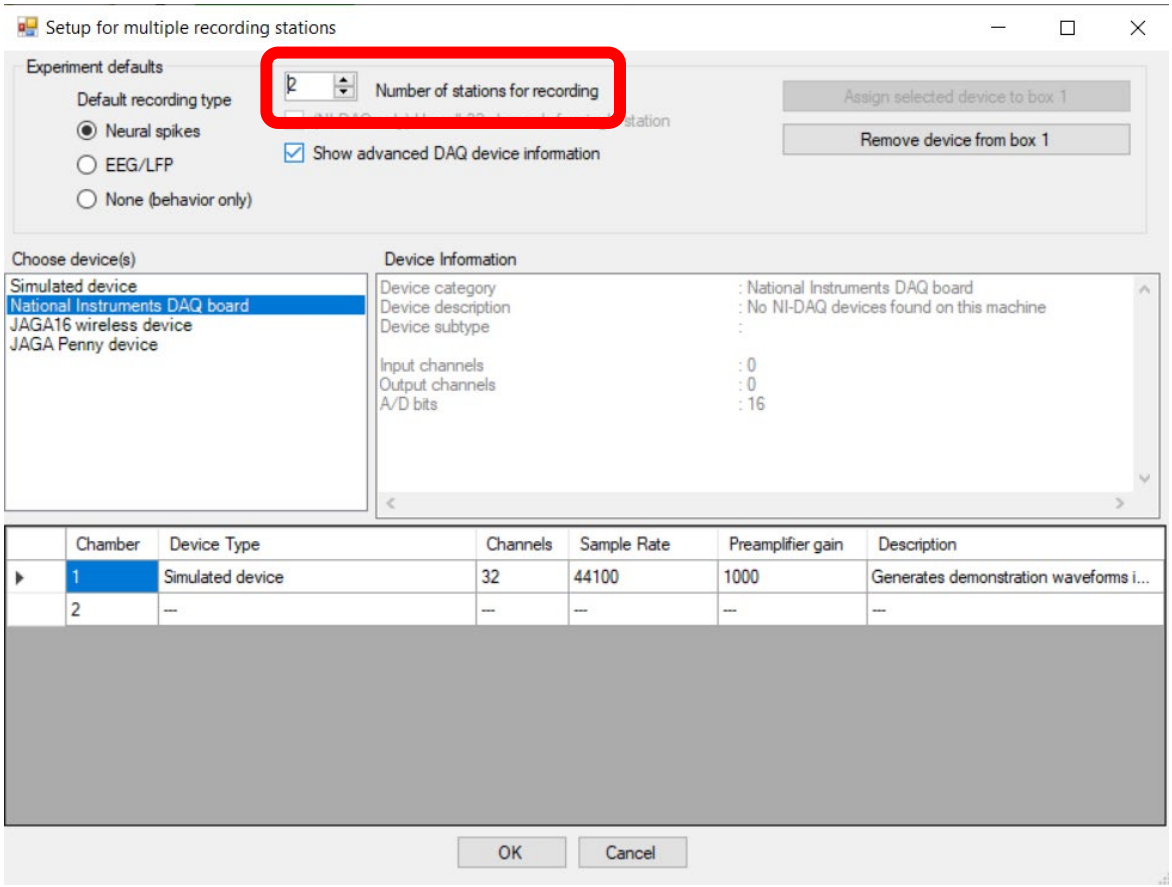


Please choose the experiment type you will perform most often – either "Neural Spikes" (default) or EEG/LFP. You can change this later at any time from the main window under "**Tools→Setup DAQ (single subject)**".

Next, select "National Instruments Device". If this option is greyed out, it means the DAQ board or NIDAQmx drivers did not install correctly. Please check the installation, and if you still have trouble, email me at tom@neurosysllc.com. If you are evaluating the software without hardware, select "Simulated Device". Click "OK", and the main window will now appear:



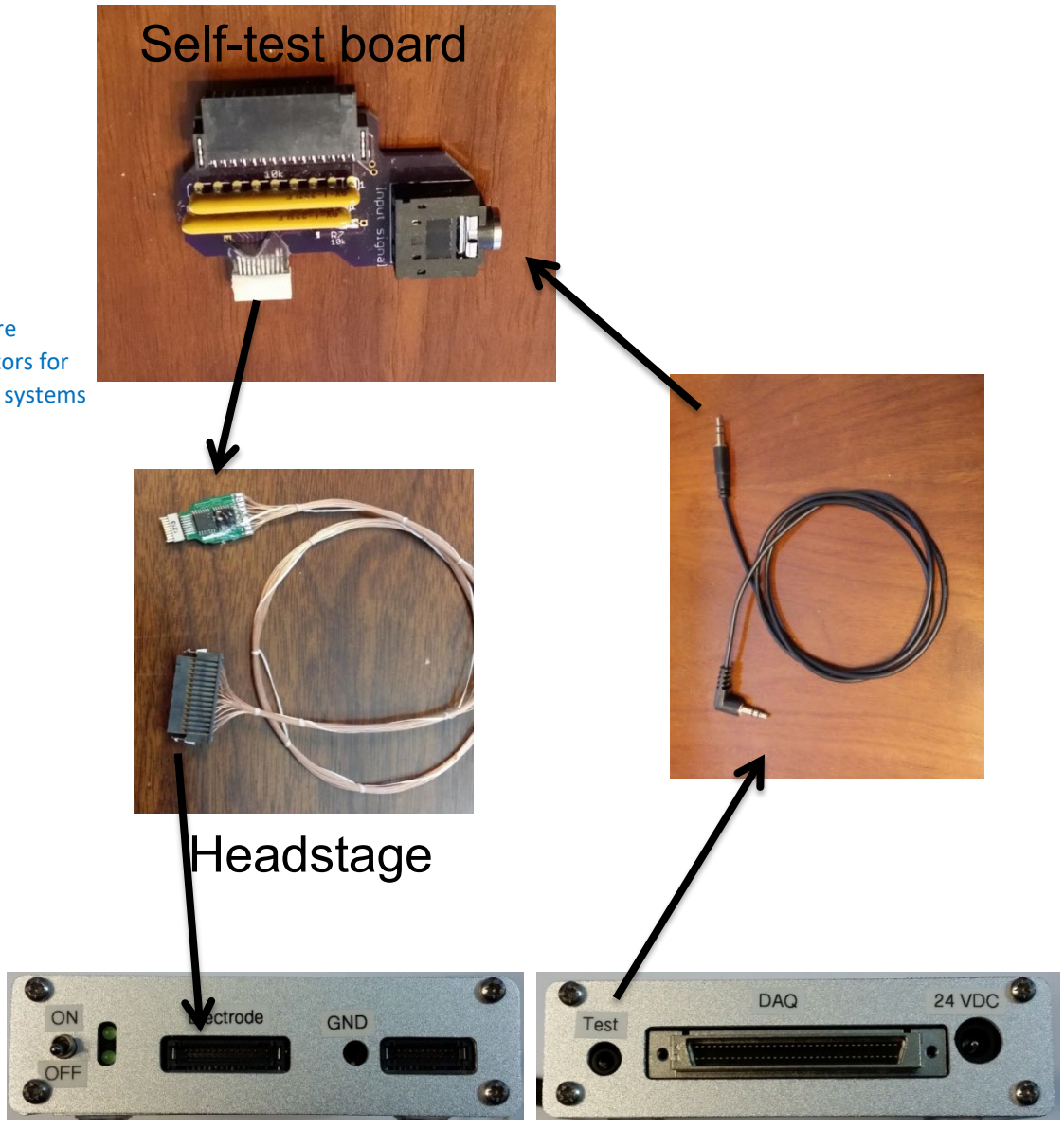
If you installed more than one NI-DAQ board into the same computer, then only one DAQ board will be configured initially. You can continue to evaluate the software using the single board. At any time, you can configure additional boards by clicking “Tools→Setup DAQ (multiple subjects)” from the top of the main window. This will bring up the dialog box below. Then, next to “Number of stations for recording”, select the number of boards, then assign each device to the appropriate box:



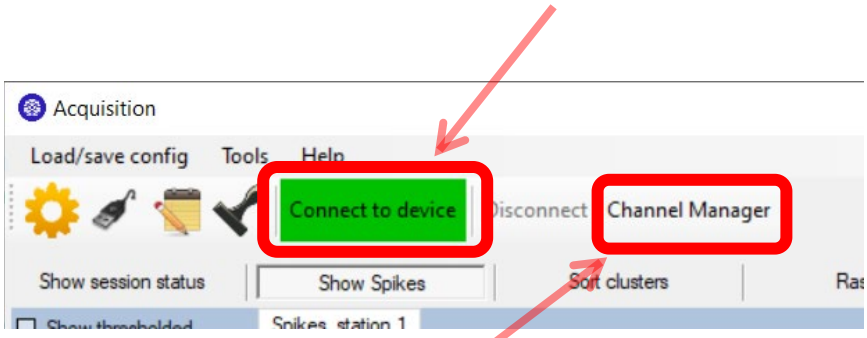
Chapter 5: System self-test

Please connect all components in the sequence indicated by the black arrows below. If you are evaluating the software using the simulated device, you can skip this step:

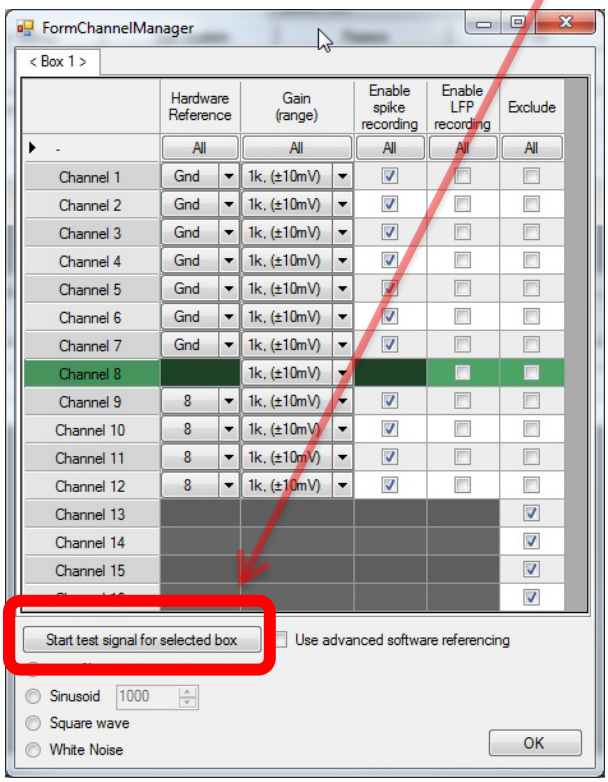
White connectors are “Omnetics” connectors for mouse systems. Rat systems use larger “Samtec” connectors.



Now click the big green button labeled “Connect to device”:



If you are evaluating the software using the simulated device, skip to the next page. Otherwise, click the “Channel Manager” button, and then “Start test signal for selected box”, which will output a test waveform from the 3.5mm audio connector on the back of the amplifier:



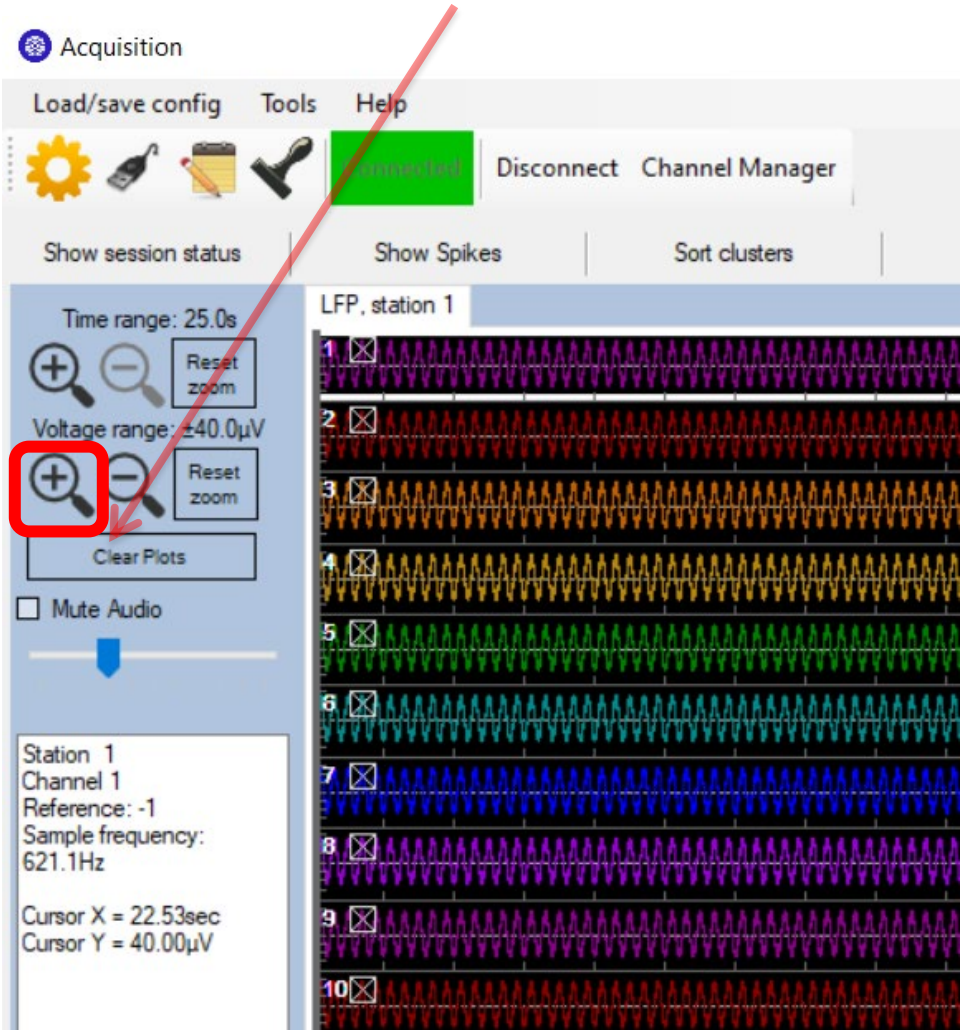
You should now see signals moving across the screen in real time. If your experiment type is “Neural Spikes”, you will see 16 signal windows arranged in 4 rows of 4:

Note that each channel window is numbered, and has a distinct color. If you selected “EEG/LFP” as your experiment type, the 16 windows will be stacked on top of each other like this:

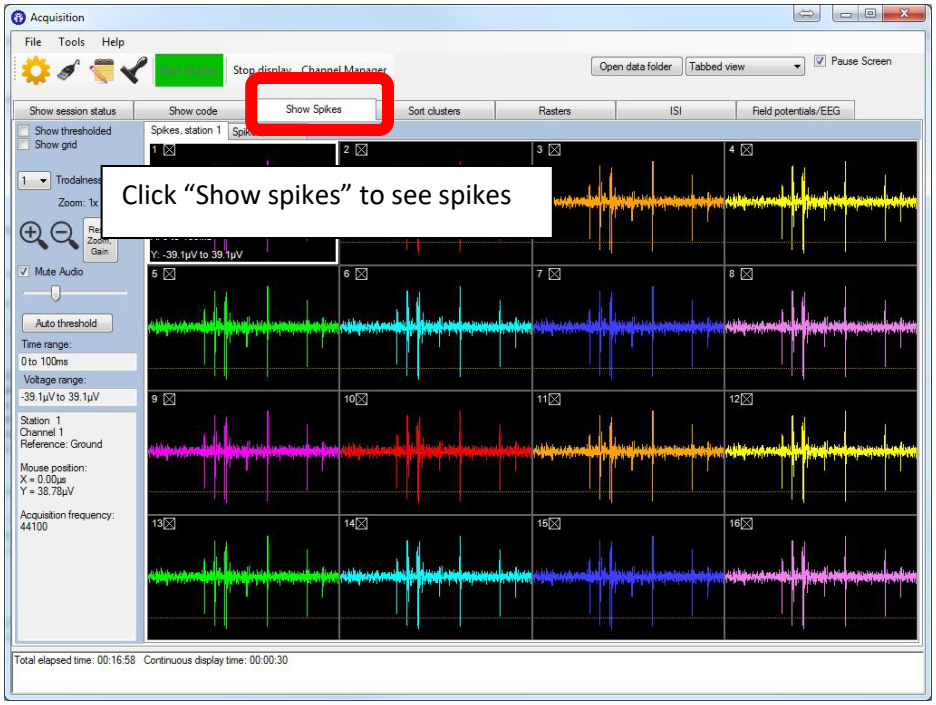
If you don't see colored traces, please check that everything is plugged in. Also check the status window at the bottom of the main window, which will turn red and report an error if the amplifier is not detected, or is powered off.

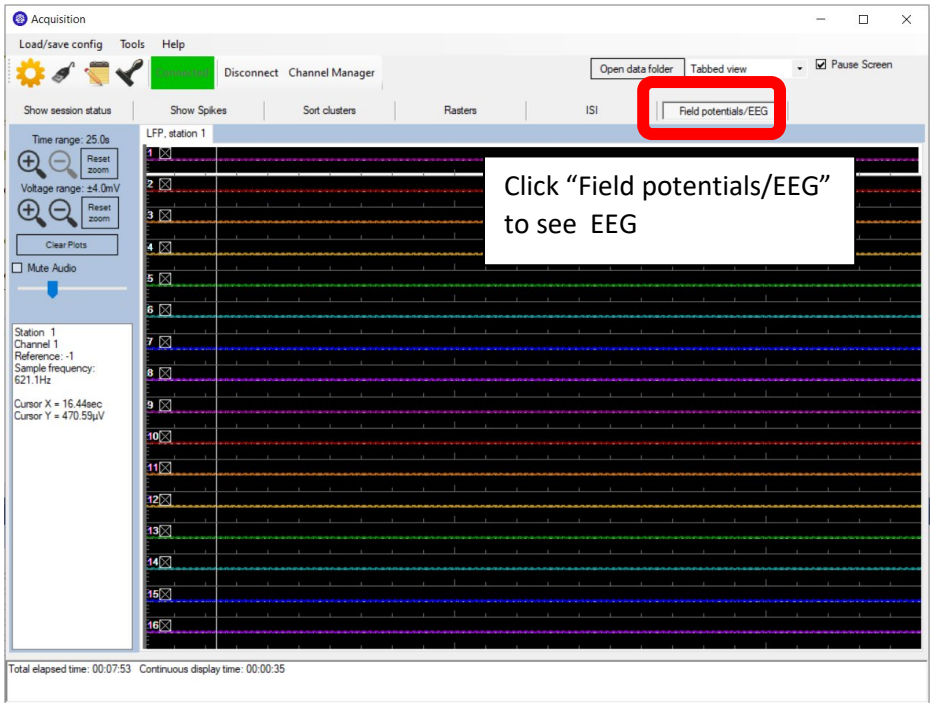
Status window. Will turn red and report error if preamplifier is off or unplugged.

Once you see traces on the screen, you can zoom in or out using the magnifying glass buttons. For example, in the Field Potentials/EEG window, the test signal is only a few millivolts, and you will need to click the "Voltage range" magnifying glass to see it (will look like a 5Hz sinusoid):



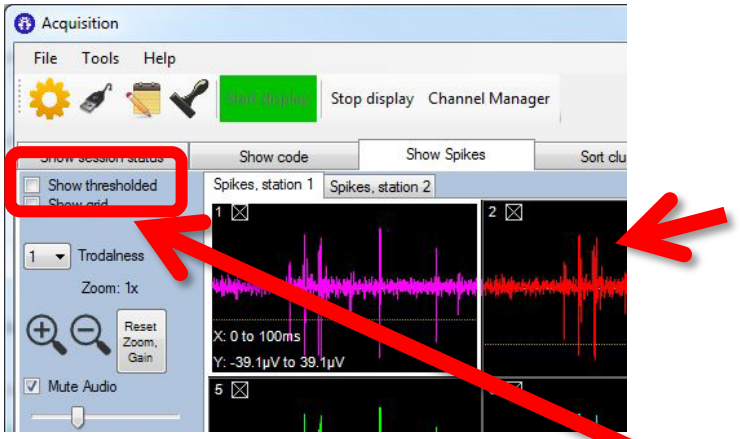
Note that the spikes and EEG are actually the same signals, but filtered differently. Spike signals are high-pass filtered, while EEG signals are low-pass filtered. And if you selected Spike mode, you can still click on the Field potentials/EEG tab to view field potentials (but they will appear grey). Conversely, if you selected EEG/LFP recording mode, you can click “Show Spikes” to see the spike signals (but they will appear grey).





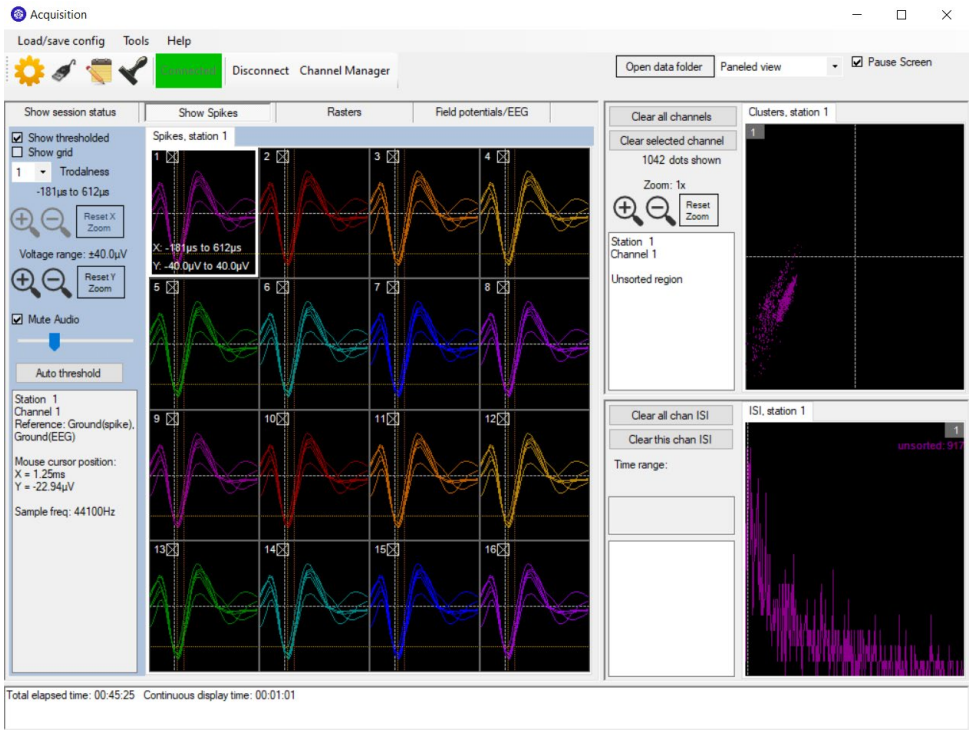
Chapter 6: Navigating the interface.

Spike windows and EEG windows both act like oscilloscopes showing data in real time. The spike window shows 50ms of data, while the EEG windows show 25 seconds. You can **double-click** any window to show that channel at full screen-size, then double-clicking again to go back.

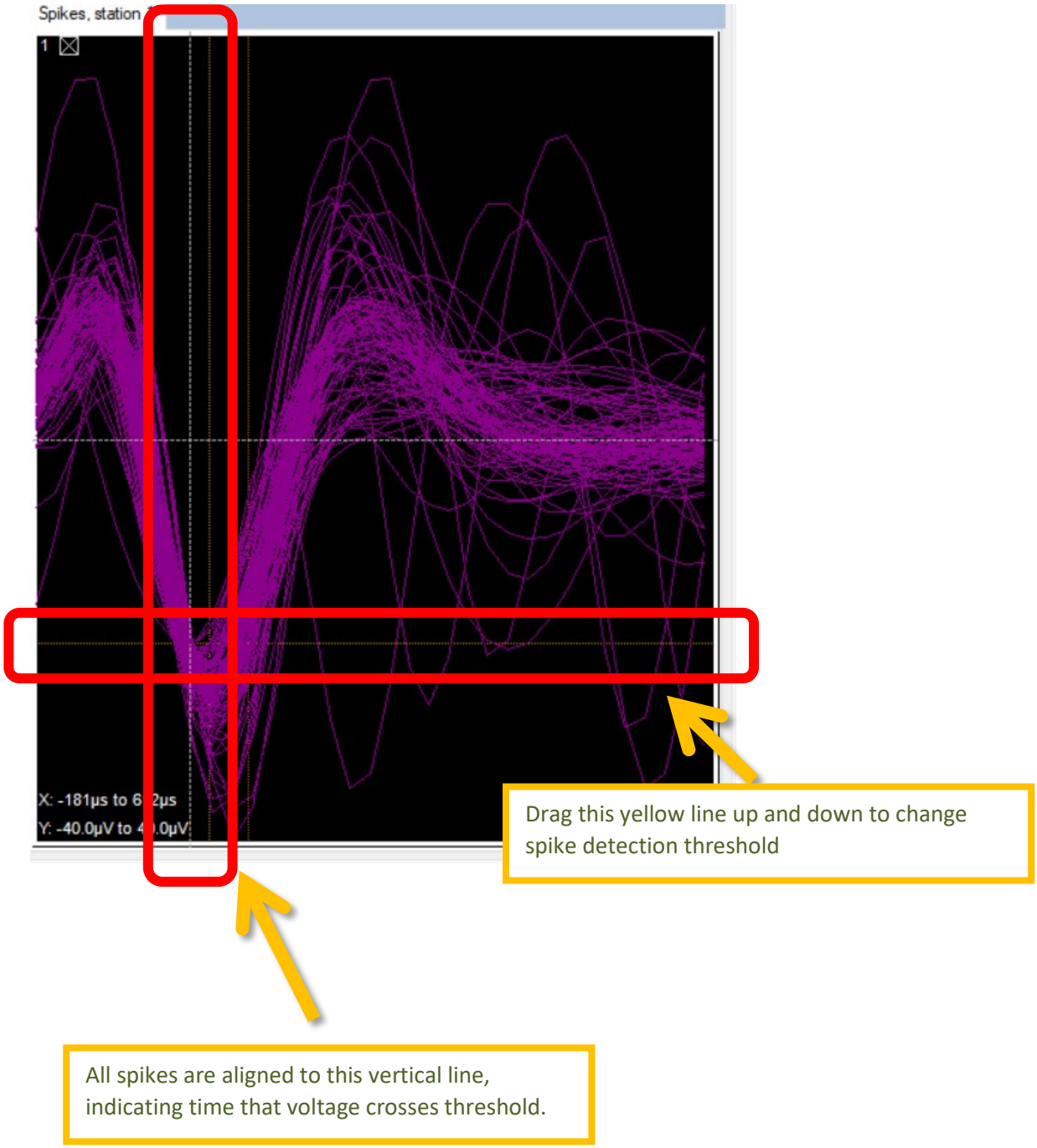


Double-click any panel to toggle full screen

In the spike window, you can also show detected spikes by clicking the "**Show thresholded**" checkbox. This will cause the windows to show only waveforms that exceed a threshold voltage:

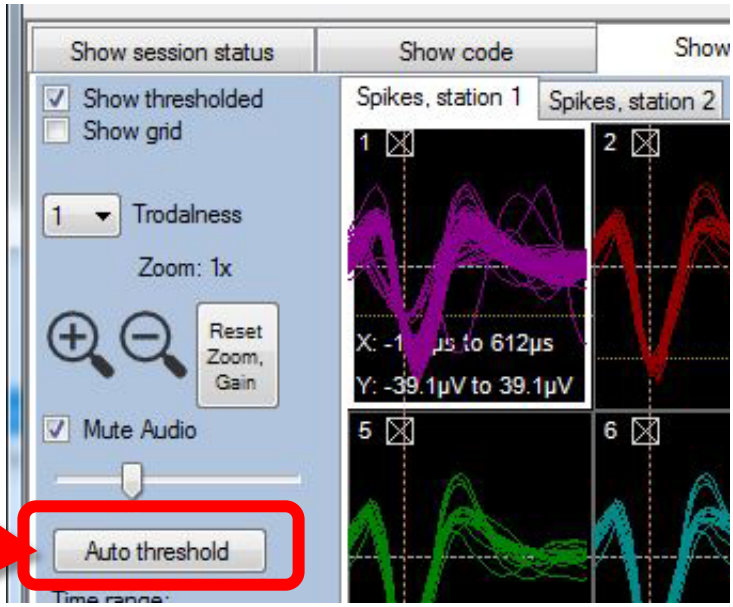


In this “thresholded” mode, the window will show only 800µs snippets of waveforms, with 200µs being prior to when the threshold was crossed, and 600µs being afterward. (The exact duration may vary, since they are integer multiples of the sample period, which is 64µs for NI boards, and 22.7µs for simulated signals.) The detection threshold voltage is set by a *horizontal yellow line* (see below) that you can drag up and down with the mouse.



You can also set thresholds automatically by ***clicking*** the "**Auto threshold**" button and then ***holding this button down for at least two seconds***. The thresholds will dynamically adjust while the button is down, and once you let go, the thresholds will lock in place. Each channel's thresholds will be computed independently.

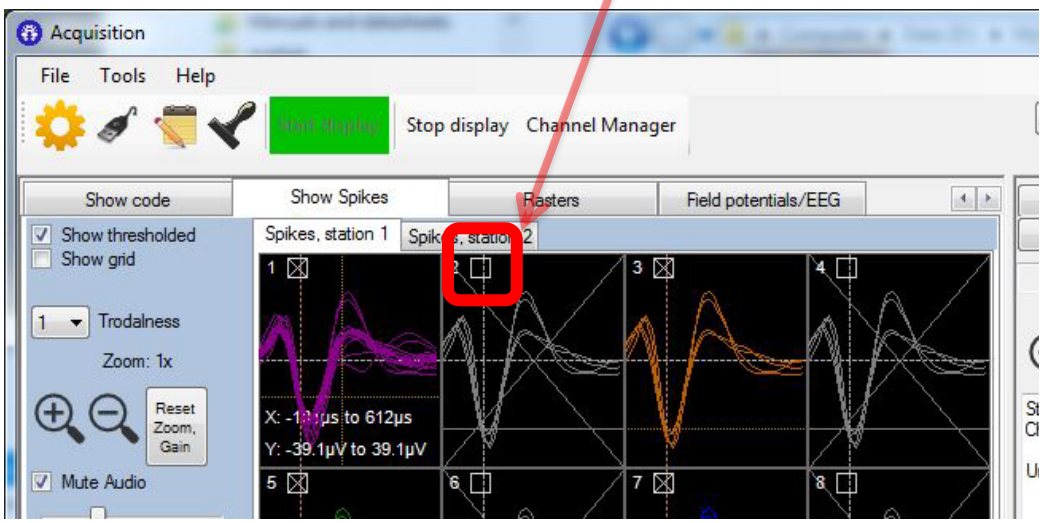
Click and **hold for 2 seconds** to automatically set threshold for each channel.

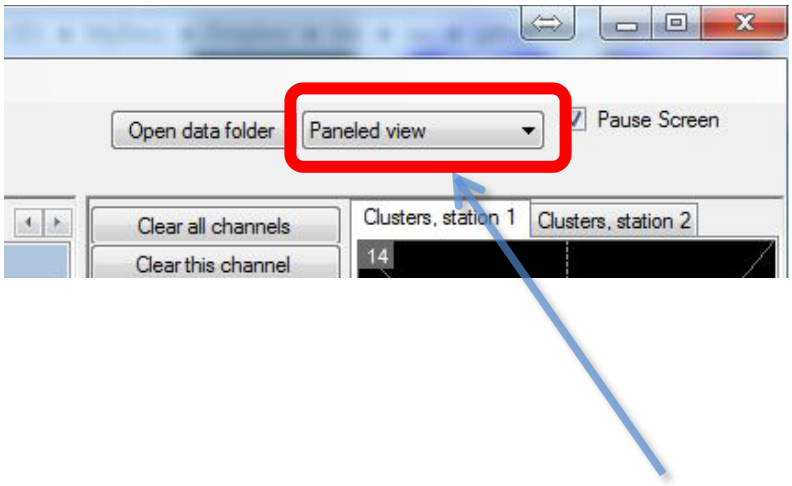


Note that when viewing detected spikes, the screen will automatically switch to "paneled" mode, which simultaneously shows "cluster" and "interspike interval" plots, which are helpful for real-time spike sorting. This is explained in more detail in the chapter on spike sorting.

Enable/disable recording for each channel:

Although the system has 16 channels, you might not want to record signals from all channels in every experiment. To save disk space, you can disable recording undesired channels by unchecking the square box in the top left of each channel window. Disabled channels will turn grey, as shown below:

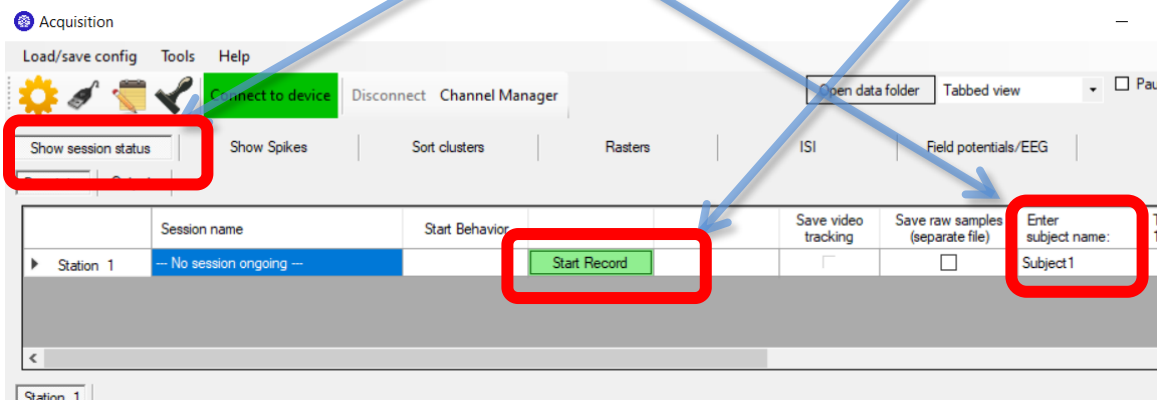




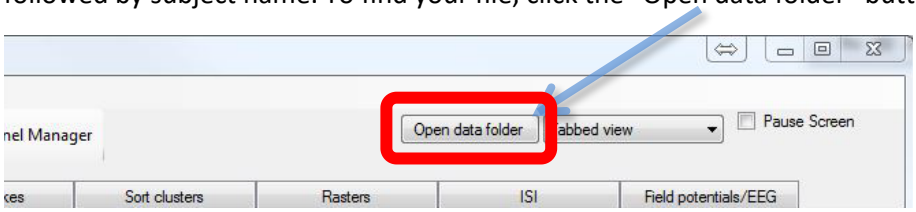
As mentioned above, selecting thresholded mode automatically switches the window from "tabbed" mode to "paneled" mode. You can manually switch between these modes using the drop-down menu at the top right of the screen.

Chapter 7: Recording a session

To record signals to your hard drive, make sure you have already clicked “Connect to device” (the big green button at top). Then click “Show session status”. You can enter an optional subject name in the column titled “Enter subject name”. When ready, click “Start Record”:

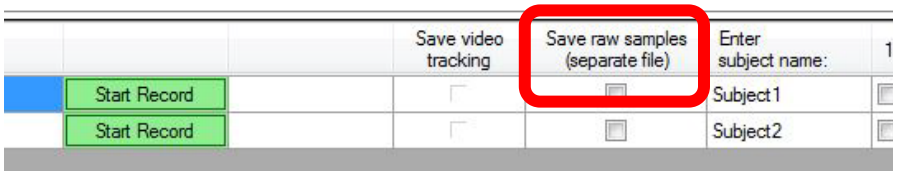


If you have more than one DAQ board configured, you will see one line for each recording station, and you can start/stop each station independently. Signals will record to disk until you click the stop button to the right of the start button. There will be two recorded files, both having names consisting of the year, month, day, hour and minute that the session began, followed by subject name. To find your file, click the “Open data folder” button:



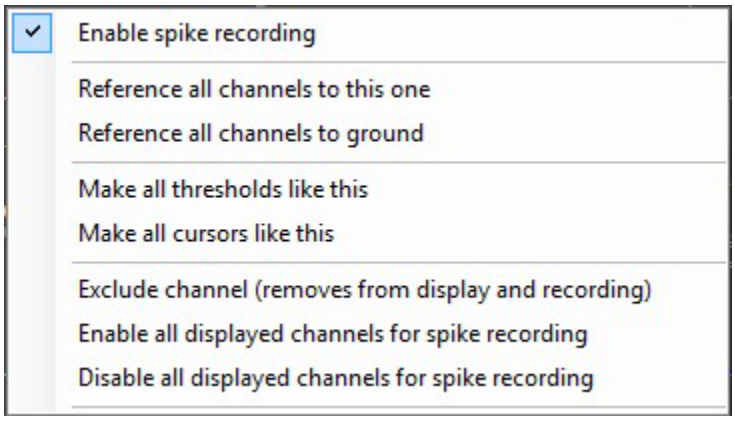
Of these two files, one is a text session log file, and the second is a Plexon PLX format file containing the actual ephys data. The latter can be read using the NeuroSorter program you can download from www.neurosysllc.com/Software.html

By default, ONLY DETECTED SPIKES are saved to disk, i.e. only the 800µs waveforms visible in the spike window above. If you want to record *all* samples, check the “Save raw samples” checkbox. However, this will result in HUGE files, and is not generally recommended.



Controlling what signals get recorded:

If you click the right mouse button on any spike window, you will see this menu, which controls how data is acquired and saved:



Enable spike recording: This option has the same effect as checking/unchecking the box in the upper left of each window.

Reference all channels to this one: This allows you to choose one wire whose voltage is subtracted from all other wires. Only the subtracted voltages are saved to disk. This greatly reduces noise, which is often present equally across multiple wires.

Reference all channels to ground: This is the default state when program is first launched.

Make all thresholds like this: This option makes all channels in this box have the same spike-detection threshold voltage as the channel you just clicked.

Make all cursors like this: This refers to the vertical cursors used to calculate the voltages in the 2-D cluster plots. For more information, see the chapter on spike sorting.

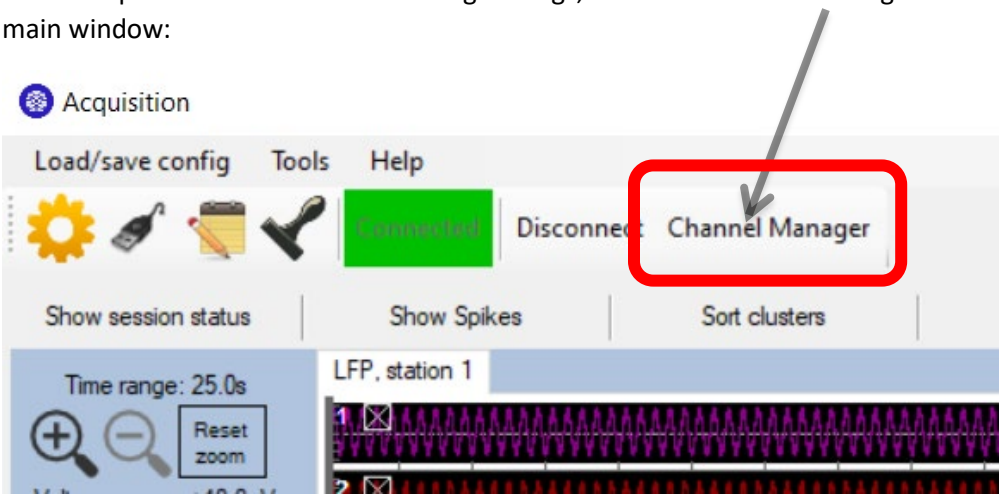
Exclude channel: This disables recording for this channel, and also hides it entirely from view. Use this to reduce screen clutter from channels you know in advance will not be useful. To "un-exclude" a previously excluded channel, use the Channel Manager (see next page).

Enable all displayed channels for spike recording: This is equivalent to checking each checkbox in the upper left of each channel window. This will cause subsequent sessions to save all channels to disk.

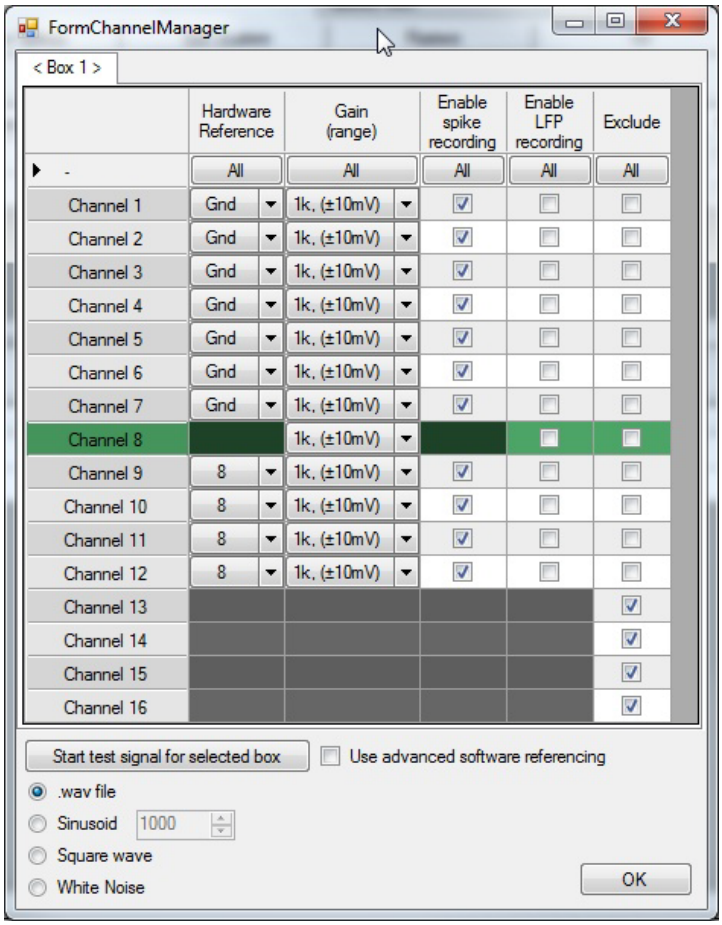
Disable all displayed channels for spike recording: This is equivalent to unchecking the checkboxes in each channel window. No channels will be saved.

Channel manager:

For more precise control over recording settings, click the “Channel Manager” button on the main window:



This gives you the following window:



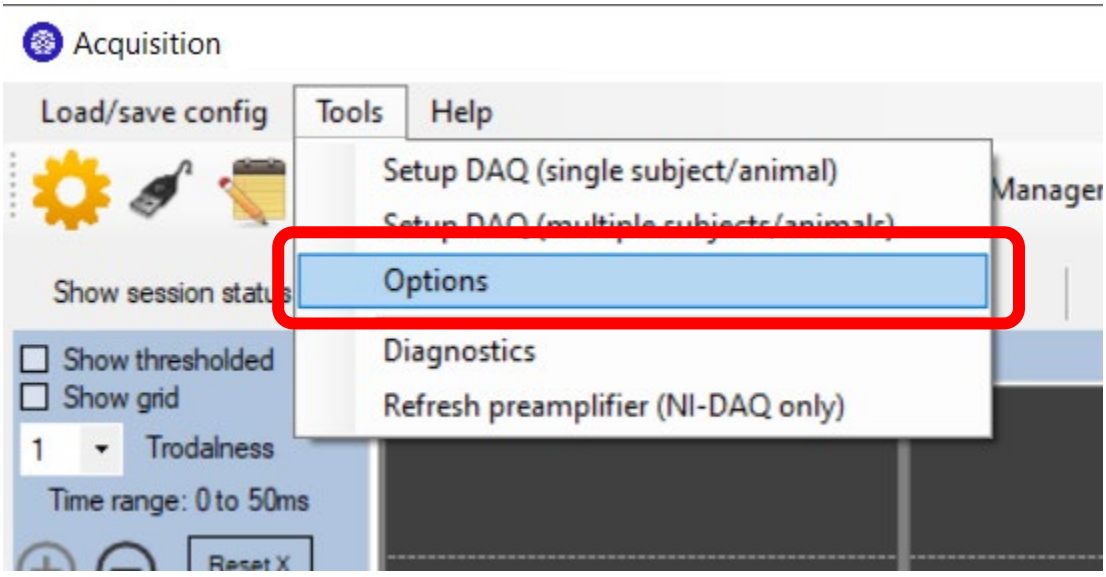
If you have more than one recording chamber configured, you will see one tab for each chamber, with the tabs being named “<Box1>”, “<Box2>”, etc. Unlike the menus above, the channel manager allows you to select a separate reference for each channel. In the above example, channels 1-7 are referenced to ground, while channels 9-12 are referenced to channel 8. The differential subtraction is performed prior to amplification, which allows for excellent common mode rejection.

Any channel used as a reference will be highlighted in green.

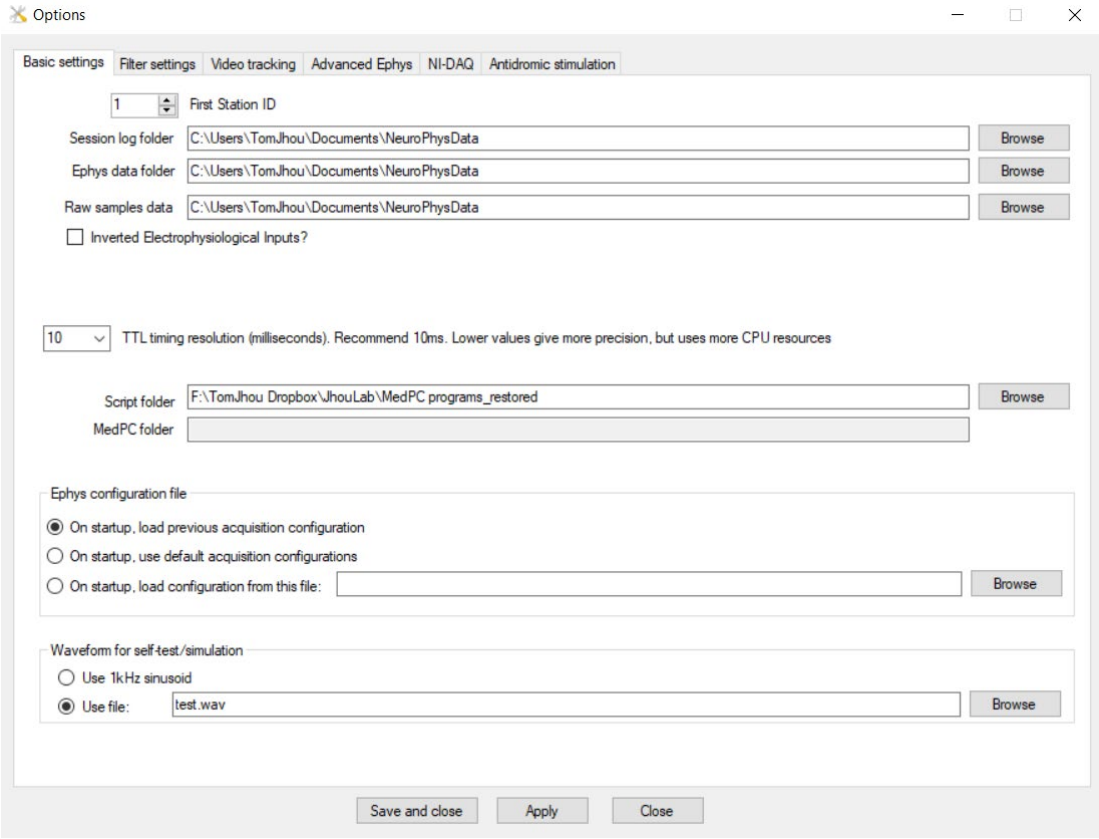
You can also exclude some channels from display and recording by checking the very last column “Exclude”. These channels will not be displayed, reducing CPU load. This also prevents users from accidentally recording a channel or using it as a reference. This is recommended if some wires are permanently unusable, either because they are not connected, or are damaged.

Changing default data folders.

By default, all recorded waveforms are saved to in the folder `NeuroPhysData`, under “`My Documents`”. You can change the folders by clicking “Tools”, “Options”.

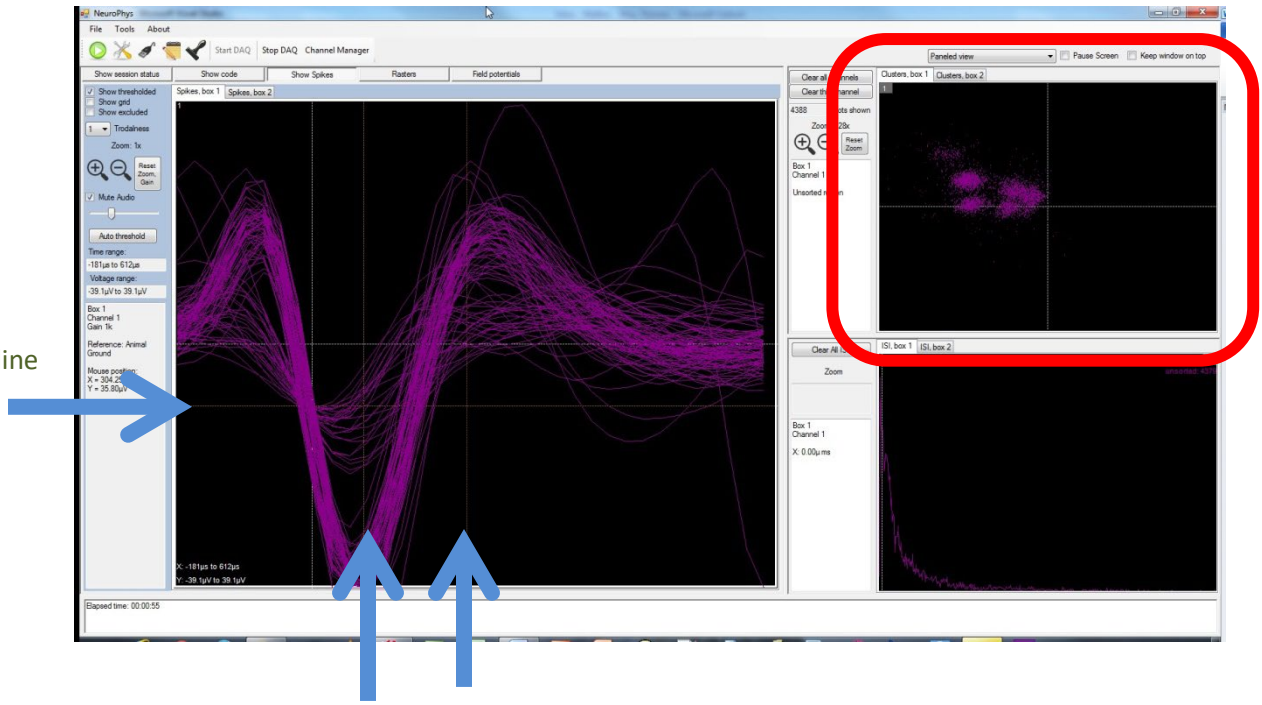


This then brings up the following dialog for choosing folders for session logs, and ephys recordings:



Chapter 7: Real-time spike sorting

This system has the ability to "sort" and classify spikes in real time. To demonstrate these capabilities, you can start a self-test session as described above, then check the "Show Thresholded" checkbox. You should see a cluster display in the upper right:



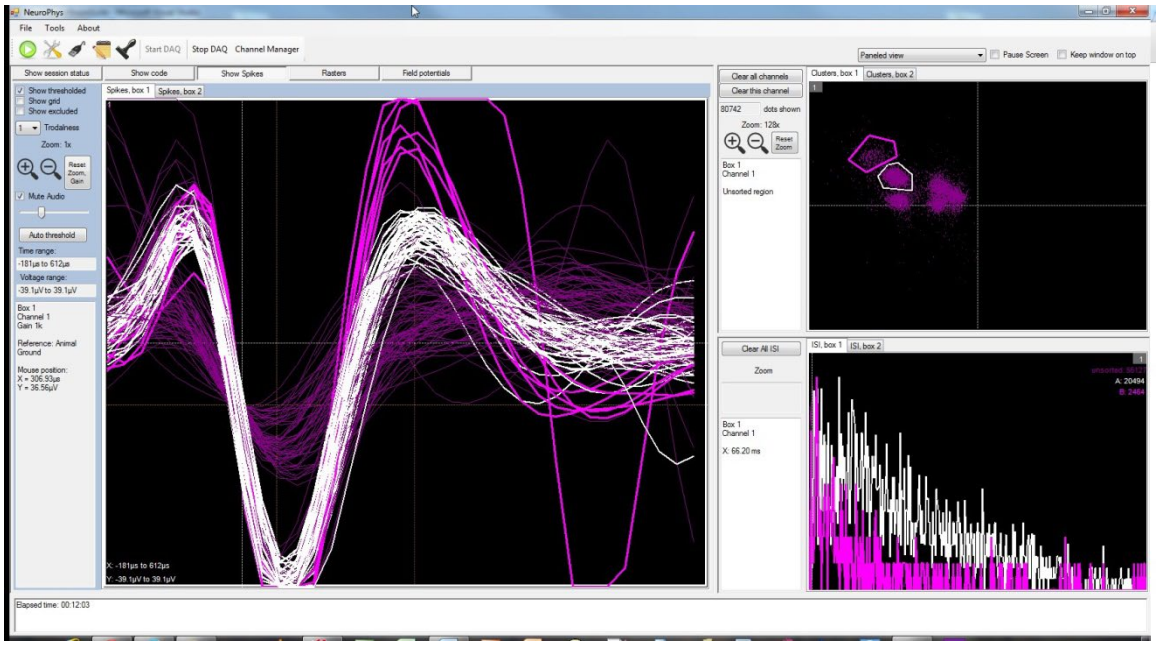
In the example above, I have made channel 1 full-screen by double clicking on it, so that it is the only spike channel visible. Note that there are three yellow dashed lines that can each be dragged with the mouse:

1. One horizontal voltage threshold cursor
2. Two vertical time cursors

The cluster plot in the top right shows one dot for each detected spike. The X coordinate of each dot is the *voltages at that spike waveform intersects time cursor #1.* Its Y coordinate *is the*

voltage at which the same wave intersects time cursor #2. In the example above, dots are mostly in the upper-left quadrant, because most waveforms intersect time cursor #1 at a negative voltage, and time cursor #2 at a positive voltage.

Note that I have placed time cursors #1 and #2 where the waveforms reach their approximate minimum and maximum voltages, respectively. This causes the dots to be as far as possible from background noise. Note how the dots segregate into multiple distinct clusters. These are often different neurons. You can click and drag the mouse to outline a polygon that encircles one of the clusters, then double-click to finish the polygon. This will cause all the waveforms corresponding to that cluster to display in a distinct color. You can place multiple polygons, and each will show up as a different color, with successive clusters designated by letters “a”, “b”, “c”, etc. This designation is saved along with the spike timestamp and waveform during a recording session. The result should look something like the window below where I have outlined two clusters:



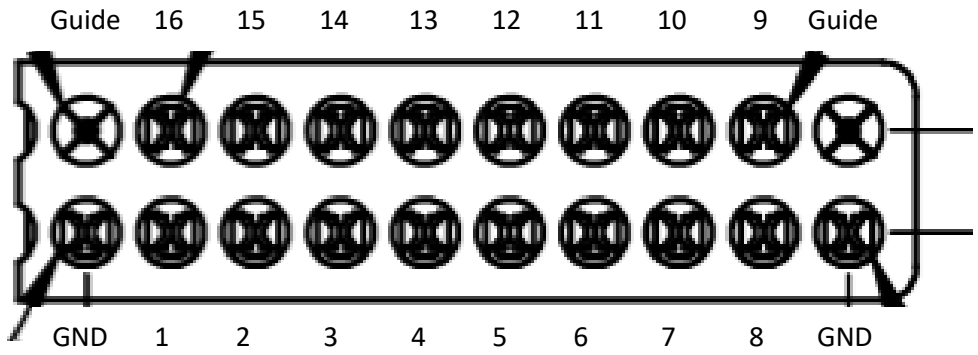
When you complete a session, click on the red “Stop” button under the “Session Status” tab. You can now view your data file in the NeuroSorter program you downloaded (next chapter).

Chapter 8: ELECTRODE FABRICATION

Mouse system headstages mate to one of the following Omnetics connectors:

1. A79014, NPD-18-DD-GS (18 signal pins, 6 guide holes, straight tails)
2. A79016, NPD-18-AA-GS (18 signal pins, 6 guide holes, horizontal surface mount)
3. A79018, NPD-18-VV-GS (18 signal pins, 6 guide holes, vertical surface mount)
4. A79038, NPD-18-DD-GS (18 signal pins, 2 guide holes, straight tails)
5. A79040, NPD-18-AA-GS (18 signal pins, 2 guide holes, horizontal surface mount)
6. A79042, NPD-18-VV-GS (18 signal pins, 2 guide holes, vertical surface mount)

When viewed from above the rat (facing down toward the rat's head), the pins are arranged as follows:



Chapter 9: Matlab integration

If you are controlling behavior from Matlab®, here is how you can send behavioral event timestamps to NeuroPhys, which will be recorded alongside electrophysiological data:

Before starting a recording session, you need to run one line of code, either from the Matlab prompt, or within a *.m file. If you are running 64-bit Windows (which will be the case for most recently purchased computers), run the following:

```
>> loadlibrary('c:\Windows\SysWow64\NS_Library.dll', 'c:\Program Files
(x86)\NeurosysLLC\NeuroPhys\NS_Library.h')
```

If you have an older computer running 32-bit Windows, run the following command instead:

```
>> loadlibrary('c:\Windows\System32\NS_Library.dll', 'c:\Program
Files\NeurosysLLC\NeuroPhys\NS_Library.h')
```

You may be prompted to selected a compiler by typing “mex –setup”. The default, “lcc-win32” compiler will work fine.

You only have to run the above command *once* per Matlab session. It loads the shared DLL that makes a number of functions accessible to Matlab’s “Calllib” function. For example:

```
Calllib('NS_Library', 'NS_StartRecording', boxNumber)
```

This function allows Matlab to initiate a recording session in NeuroPhys. “boxNumber” is an integer referring to the operant chamber number (the lowest numbered box will be “1”). If you have only one chamber, this will most likely be “1”.

```
Calllib('NS_Library', 'NS_SendEvent', boxNumber, eventNumber, 0)
```

This function sends a timestamped event to NeuroPhys, which will be recorded alongside any ongoing electrophysiological data. The timestamp will be calculated at the exact instant this

function is called, and should be accurate to well under a millisecond. *boxNumber* is the operant chamber index. *eventNumber* is a user-designated ID number between 0 and 249. Typically, one would assign a unique ID to specific events, e.g. a leverpress could be ID 1, a nosepoke would be ID 2, etc. There is a final argument that is reserved for future expanded features, and for now should always be "0".

```
Calllib('NS_Library', 'NS_StopBox', boxNumber)
```

The "NS_StopBox" command will stop a recording session in the specified operant chamber.

```
Calllib('NS_Library', 'NS_StartRecordingWideband', boxNumber)
```

This allows Matlab to start recording wideband signals, i.e. it will record the raw voltages in a separate file at the full sampling rate available. This will generate very large files that retains all possible information that can be recorded. This is useful if you don't know exactly how you will analyze your data, and wish to process the session off-line at a later time.

Chapter 10: Med-PC® integration

Probably the best way to illustrate Med-PC® integration is with a sample MPC file. After installing NeuroPhys, this file will appear in "..\My Documents\NeuroPhysData", along with a couple of sample data files:

```
\
\  RecordingTest.MPC
\
\  Simple test file to show communication between MedPC and NeuroPhys.
\

^LeftLever = 1          \ Left lever is output 1. If your chamber is
wired differently, you should change this.


s.s.1,


s1,
  0": ~sendeventname(BOX, 1, 'ProgramStart');~;
      ~sendeventname(BOX, 2, 'LeverPress');~; ---> S2


S2,
   #start: ~startRecording(BOX);~;
           ~sendtext(BOX, 'Program starting' + #13 + #10);~;
           ~sendevent(BOX, 1, 0);~; ---> S3


S3,
   #R^LeftLever: ~sendevent(BOX, 2, 0);~; ---> S4
```

```
S4,
    5": ~stopBox(BOX);~; ---> StopAbortFlush
```

Copy the above file from "My Documents\NeuroPhysData" into "C:\MED-PC IV\MPC". Then compile it using TRANS®, and run it from Med-PC®. Make sure that NeuroPhys is already running before you run it.

The command "StartRecording" will cause NeuroPhys to start recording spikes. This program will wait for a single press on the left lever, and then quite 5 seconds later.

Note that all communication with NeuroPhys occurs via functions called from PASCAL, so all such functions begin and end with a tilde symbol.

The command "sendevent" sends an event with the specified ID number (1 or 2 in the above example). The command "sendeventname" is optional, and is used to send a user-friendly name to NeuroPhys, which will then write it into the PLX file header. When you later open your PLX file in the spike sorter, or in Neuroexplorer, these user-friendly names will appear, and will make your life a lot easier.

Just before the Med-PC program ends, call "StopBox" to stop the recording session. If you forget to include this line, NeuroPhys will keep recording, and you will have to stop it manually (the red button on the box status tab).

Chapter 11: Reading PLX files in MATLAB or C/C++

NeuroPhys records data in the same format as used by Plexon Inc. These files have the extension “PLX”.

NeuroPhys includes an example code snippet that will read PLX files in either Matlab or C/C++. This code example should be enough to get you started. This code is based on that supplied by Plexon in their “OmniPlex and MAP Offline SDK Bundle”, which can be downloaded from http://www.plexon.com/software-downloads .